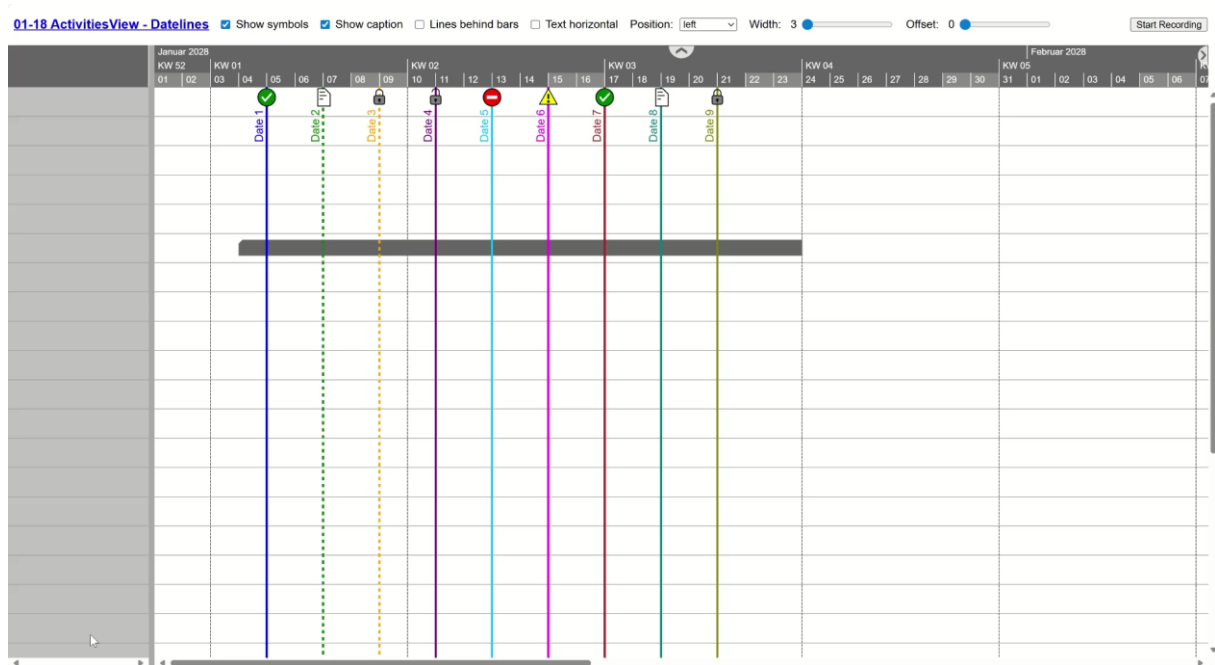


90-02 All views – Basics – Date lines

This example demonstrates how date lines can be used. In addition to their visual design, date lines can include a label and a symbol. It is also possible to configure whether they are movable and whether they should appear in front of or behind the bars.



Like all other objects, each date line has a unique ID for identification. The most important property is **PointInTime**, which defines the position of the date line. The array of date lines is added to the **vsWidget** using the **addDateLines** method.

If symbols are required, they must also be defined and added to the **vsWidget**.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <link href="../../VSWidget/nwaf-apptools.min.css" rel="stylesheet"/>
  <link href="../../VSWidget/nwaf-table.min.css" rel="stylesheet"/>
  <link href="../../VSWidget/nwaf-gantt.min.css" rel="stylesheet"/>
  <link href="../../VSWidget/nwaf-rab.min.css" rel="stylesheet"/>
  <script src="https://cdn.jsdelivr.net/npm/hammerjs@2.0.8/hammer.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/d3@7.9.0/dist/d3.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/tinycolor2@1.6.0/cjs/tinycolor.min.js"></script>
  <script src="../../VSWidget/nwaf-apptools.min.js"></script>
  <script src="../../VSWidget/nwaf-table.min.js"></script>
  <script src="../../VSWidget/nwaf-gantt.min.js"></script>
  <script src="../../VSWidget/nwaf-rab.min.js"></script>
  <script src="../../LicenseKey.js"></script>
  <script src="Miscellaneous.js"></script>
  <style>
    .slider-output {
      display: inline-block;
      min-width: 2ch; /* Platz für mindestens zwei Ziffern (z.B. 10) */
      text-align: right;
    }
  </style>
</head>
<body>
  <div id="toolbar">
    <input style="margin-left: 12px" id="showSymbol" type="checkbox" checked/>
    <label for="showSymbol">Show symbols</label>
    <input style="margin-left: 12px" id="showCaption" type="checkbox" checked/>
    <label for="showCaption">Show caption</label>
```

```

<input style="margin-left: 12px" id="linesBehindBars" type="checkbox"/>
<label for="linesBehindBars">Lines behind bars</label>
<input style="margin-left: 12px" id="textHorizontal" type="checkbox"/>
<label for="textHorizontal">Text horizontal</label>
<label for="position" style="margin-left: 12px">Position:&nbsp;  </label>
<select id="position" name="fields">
  <option value="1" selected>left</option>
  <option value="2">center</option>
  <option value="4">right</option>
  <option value="9">top left</option>
  <option value="10">top center</option>
  <option value="12">top right</option>
</select>
<label style="margin-left: 12px" for="slider1">
  Width:
  <span class="slider-output" id="output1">3</span>
</label>
<input type="range" id="slider1" min="1" max="15" value="1"
  data-output="output1" data-option="width"/>
<label style="margin-left: 12px" for="slider2">
  Offset:
  <span class="slider-output" id="output2">0</span>
</label>
<input type="range" id="slider2" min="0" max="24" value="0"
  data-output="output2" data-option="offset"/>
</div>
<div id="VSWidget"></div>
<script>
  Miscellaneous.ready(() => {
    const { DateLineAppearanceType, DateLineCaptionOrientation, RowDesigns } = netronic.nVSW;

    document.getElementById("showSymbol").addEventListener("click", changeDateLineDesign);
    document.getElementById("showCaption").addEventListener("click", changeDateLineDesign);
    document.getElementById("linesBehindBars").addEventListener("click", changeDateLineDesign);
    document.getElementById("textHorizontal").addEventListener("click", changeDateLineDesign);
    document.getElementById("slider1").addEventListener("input", changeDateLineProperties);
    document.getElementById("slider2").addEventListener("input", changeDateLineProperties);
    document.getElementById("position").addEventListener("change", changeCaptionPosition);

    const timeAreaStart = new Date(2028, 0, 1);
    const timeAreaEnd = new Date(2028, 3, 1);

    const options = {
      licenseKey: window.VSW_LICENSE_KEY,

      start: timeAreaStart,
      end: timeAreaEnd,

      ignoreCalendarOnActivityBarInteractions: true,
      defaultActivityExpandedRowDesign: RowDesigns.Bars,
    };

    const vsWidget = Miscellaneous.instantiateVSWidget(document.querySelector("#VSWidget"), options);

    const symbols = createSymbols();
    const dateLines = createDateLines();
    const resources = createActivities();
    vsWidget.addSymbols(symbols);
    vsWidget.addDateLines(dateLines);
    vsWidget.addActivities(resources);
    vsWidget.render();

    function createActivities() {
      const activities = [];
      for (let i = 1; i <= 30; i++) {
        const activity = { ID: `A${i}` };
        if (i === 6) {
          activity.Start = new Date(2028, 0, 4);
          activity.End = new Date(2028, 0, 24);
        }
        if (i === 10) {
          activity.AttachedDateLineIDs = ["D10"];
        }
        activities.push(activity);
      }
    }
  });

```

```

    }

    return activities;
}

function createSymbols() {
    return [
        { ID: "Symbol1", URL: "images/warning.svg" },
        { ID: "Symbol2", URL: "images/checkmark.svg" },
        { ID: "Symbol3", URL: "images/document.svg" },
        { ID: "Symbol4", URL: "images/closed.svg" },
        { ID: "Symbol5", URL: "images/opened.svg" },
        { ID: "Symbol6", URL: "images/oneWay.svg" },
    ];
}

function createDateLines() {
    const dateLines = [];
    const startDate = new Date(2028, 0, 3, 0, 0, 0);

    const colors = [
        "red",
        "blue",
        "green",
        "orange",
        "purple",
        "deepskyblue",
        "magenta",
        "brown",
        "teal",
        "olive",
    ];

    const symbols = [
        "Symbol1",
        "Symbol2",
        "Symbol3",
        "Symbol4",
        "Symbol5",
        "Symbol6",
    ];

    for (let i = 1; i < 11; i++) {
        const pointInTime = new Date(startDate);
        pointInTime.setDate(startDate.getDate() + i * 2); // +2 Tage pro Linie

        const dateline = {
            ID: `D${i}`,
            Caption: `Date ${i}`,
            CaptionColor: colors[i % colors.length],
            Draggable: true,
            Color: colors[i % colors.length],
            Width: 3,
            PointInTime: pointInTime,
            SymbolID: symbols[i % symbols.length],
            DashArray: i === 2 || i === 3 ? "5,5" : "none",
            SymbolHeight: 30,
            SymbolWidth: 30,
        };
        if (i === 10)
            dateline.AppearanceType = DateLineAppearanceType.PointerOverAttachedBar;
        dateLines.push(dateline);
    }

    return dateLines;
}

function changeDateLineProperties(event) {
    const slider = event.target;

    const value = slider.value;
    const formatted = String(value).padStart(4, " ");
    const outputId = slider.dataset.output;
    const optionKey = slider.dataset.option;

```

```

document.getElementById(outputId).innerText = formatted;

if (optionKey === "width") {
    for (let i = 0; i < dateLines.length; i++)
        dateLines[i].Width = +value;

    vsWidget.updateDateLines(dateLines);
    vsWidget.render();
}

else if (optionKey === "offset") {
    for (let i = 0; i < dateLines.length; i++)
        dateLines[i].PointInTime = new Date(2028, 0, 3 + i * 2, +value, 0, 0);

    vsWidget.updateDateLines(dateLines);
    vsWidget.render();
}
}

function changeDateLineDesign() {
    let chkBox1 = document.getElementById("showSymbol");
    if (chkBox1) {
        const symbols = [
            "Symbol1",
            "Symbol2",
            "Symbol3",
            "Symbol4",
            "Symbol5",
            "Symbol6",
        ];
        for (let i = 0; i < dateLines.length; i++)
            dateLines[i].SymbolID = chkBox1.checked ? symbols[i % symbols.length] : "";
    }

    let chkBox2 = document.getElementById("showCaption");
    if (chkBox2) {
        for (let i = 0; i < dateLines.length; i++)
            dateLines[i].Caption = chkBox2.checked ? `Date ${i + 1}` : "";
    }

    let chkBox3 = document.getElementById("linesBehindBars");
    if (chkBox3) {
        for (let i = 0; i < dateLines.length; i++)
            dateLines[i].InFrontOfBars = !chkBox3.checked;
    }

    let chkBox4 = document.getElementById("textHorizontal");
    if (chkBox4) {
        for (let i = 0; i < dateLines.length; i++)
            dateLines[i].CaptionOrientation = chkBox4.checked ? DateLineCaptionOrientation.Horizontal :
                                                                    DateLineCaptionOrientation.Vertical;
    }

    vsWidget.updateDateLines(dateLines);
    vsWidget.render();
}

function changeCaptionPosition() {
    const val = document.getElementById("position").value;

    for (let i = 0; i < dateLines.length; i++)
        dateLines[i].CaptionPosition = val;

    vsWidget.updateDateLines(dateLines);
    vsWidget.render();
}
});
</script>
</body>
</html>

```